Path Planning of Webot Satisficing Experiment —- Workspace and C-Target Construction

Min Zheng

August 5, 2017

This project considers the integrated navigation and control of an unmanned ground vehicle(UGV) deployed to visually classify multiple targets in an obstacle-populated environment. This report describes the problem setup and the contruction of the $\mathcal{C}_{Target}$.

I. WORKSPACE CONSTRUCTION

The workspace is the same as the one used for active satisficing human studies. The UGV and 30 target objects are in the same environment that consists of four rooms, denoted as the region-of-interest (ROI). Similar to the human studies, there are 30 targets, denoted as $\mathcal{T} = \{\mathcal{T}_i; i = 1, 2, ...30; \mathcal{T} \subset W\}$. The targets are considered as points. A two-dimensional representation from the webot environment was extracted, and the workspace is constructed in MatLab, as shown in the figure below (Figure 2).

The construction of workspace is built in an object oriented manner with classes of obstacles, targets and robot, each has properties and functionalities. Note that the table and the chairs are treated differently from other obstacles. Since their top surfaces are higher than the robot, only their legs will be considered as obstacles.
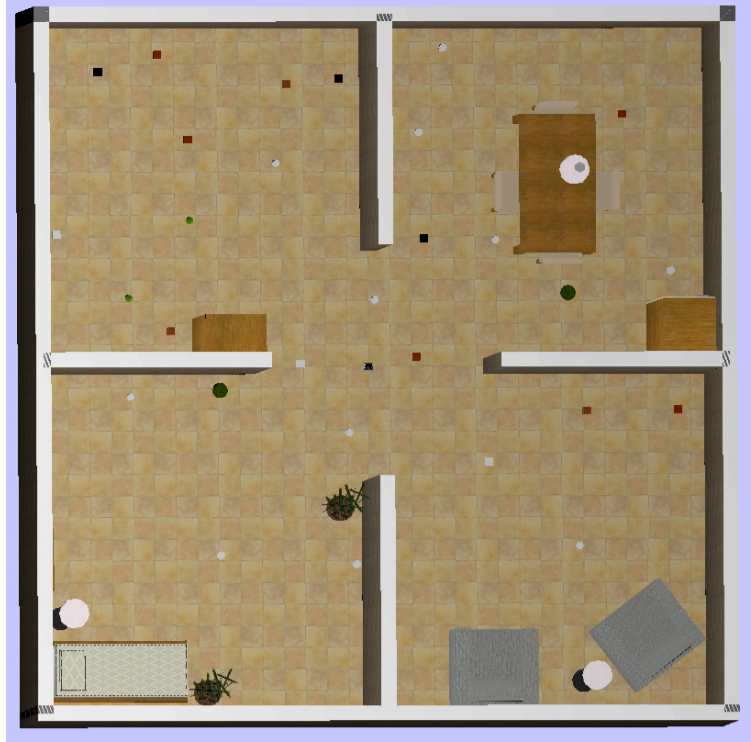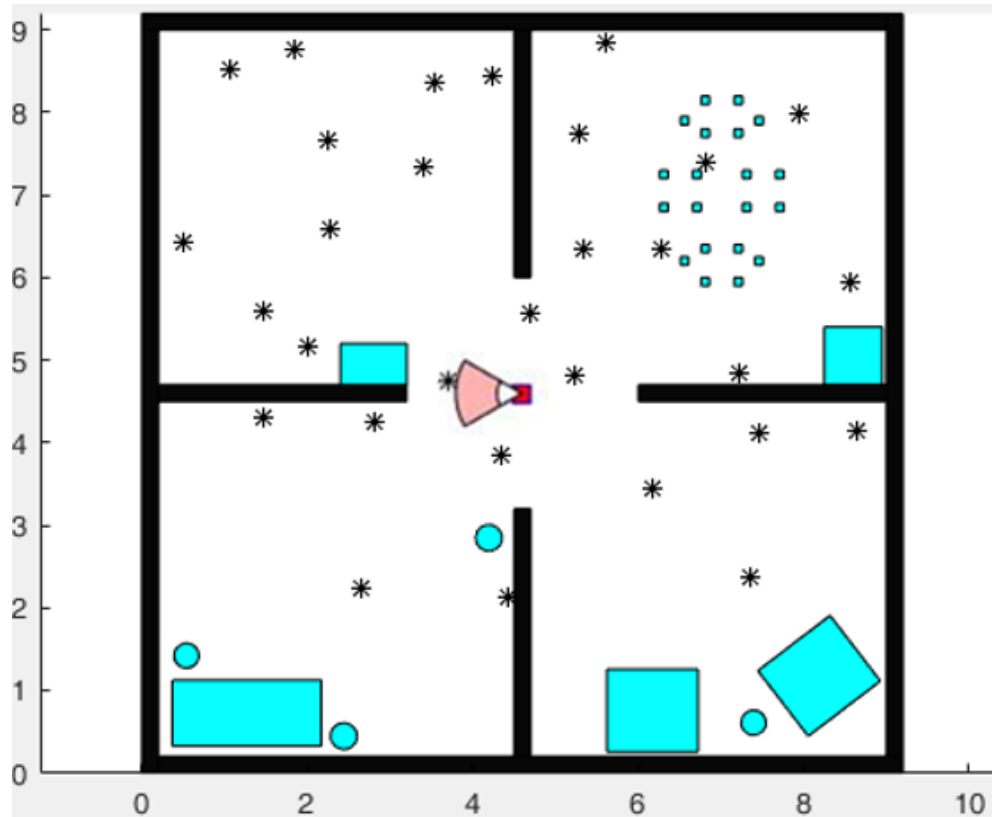
Fig. 1: Top view of the webot environment

Fig. 2: 2D representation built in MatLab. Obstacles are represented by colored regions, and targets represented by the stars. The UGV is the red square, while its FOV is the red area within the fan shape.
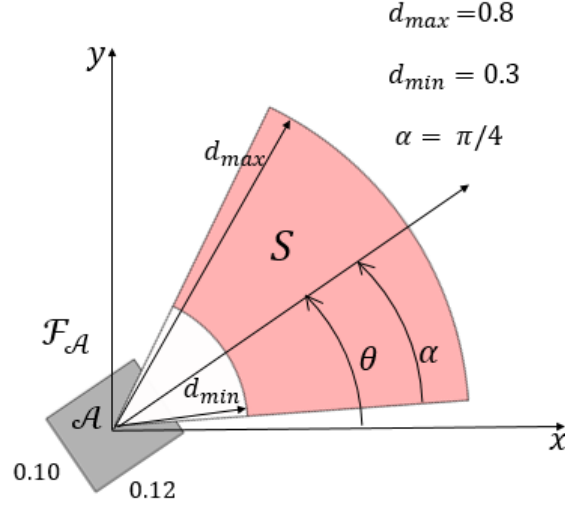
Fig. 3: Configuration and FOV of the UGV.

II. UGV's SPECS AND FOV

The UGV sensor in the webot environment has platform geometry $\mathcal{A}$, and FOV geometry $S \subset \mathbb{R}^2$. The dimension of UGV is 0.12m x 0.10m in 2D. Its translation step is 0.01m, and rotation step is $\pi/12$. The UGV configuration is described in (Figure 1). The configuration vector is defined as $q \equiv [x \quad y \quad \theta]^T$.

In order to combine with Zeyu's visual classification project, the FOV was determined based on requirement of images taken. A successful classification requires the image to be taken within a minimum and a maximum distance, denoted by d_{min} and d_{max}. In human tests, the open angle of FOV for the on board camera is $\alpha = \pi/4$. Therefore the same α is used here. Also, in the visual classification problem, Ryan took all the pictures from 0.3m. If the picture is taken less than that distance, the target object is not guaranteed to be completely within the camera view. Meanwhile, in the human tests software setup, the robot must be within 1m from the target to measure or classify. In such manner, the range of distance from target is conservatively set within $d_{min} = 0.3m$ and $d_{max} = 0.8m$. In order to better work with the visual classification, these parameters, especially d_{max} may be subject to changes due to the highly sensitive manner of visual classifier. For example, it may be affected by whether the visual classifier allows multiple objects to appear in the view.
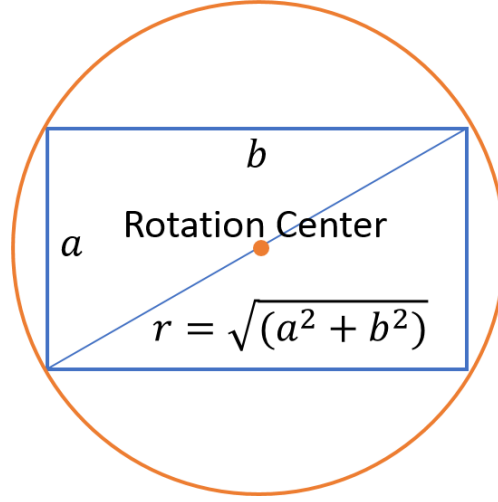
Fig. 4: The area(orange circle) swept by rotating the UGV(blue rectangle) around its center.

III. $\mathcal{C}_{free}$

Assume that the UGV can freely rotate. Thus $\alpha \in (-\pi \, , \pi)$. In x-y plane, a location is defined to be in $\mathcal{C}_{free}$ if the UGV can freely rotate around its center, which is at the given location. The locations where some subsets of all possible α are be allowed is discarded from $\mathcal{C}_{free}$, considering the problem of getting stuck. Given the geometry of the UGV, it requires that the bounding circle of the UGV (Figure 4) does not intersect with any obstacles. The result $\mathcal{C}_{free}$ is shown in Figure 5.
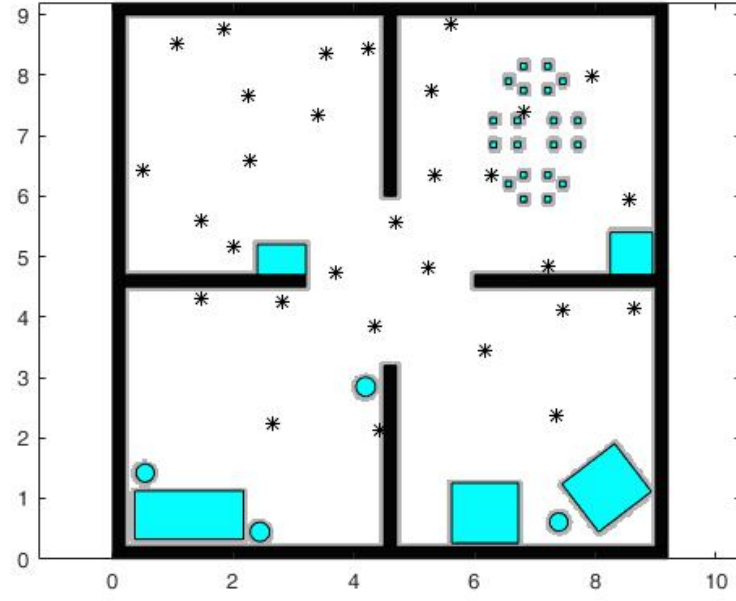
(a) $\mathcal{C}_{free}$ for the whole world map
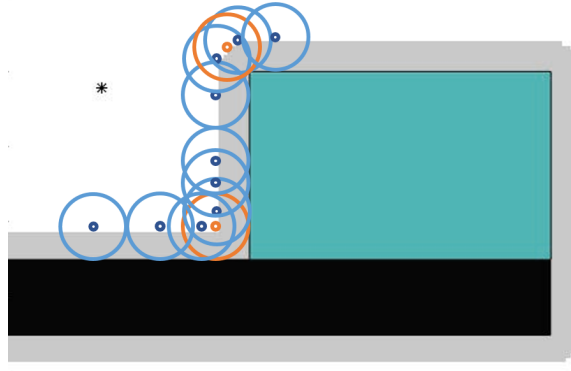
(b) zoomed-in demonstration of $\mathcal{C}_{free}$, where the center of the circle represents the UGV center location, so that the

Fig. 5: $\mathcal{C}_{free}$ of the given world map denoted by white region. Grey area is not in $\mathcal{C}_{free}$
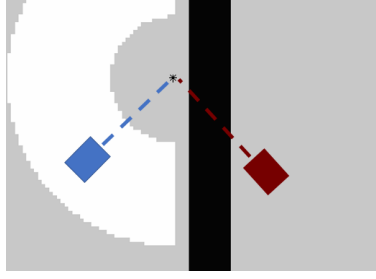
Fig. 6: Example of visibility problem handling for a target near the wall. The blue and red rectangles denote the UGV configuration. The dash line denotes the direction the robot is facing. It is also used in the visibility handling. The blue configuration is in C-Target, but the red is not since its view is blocked by the obstacle.
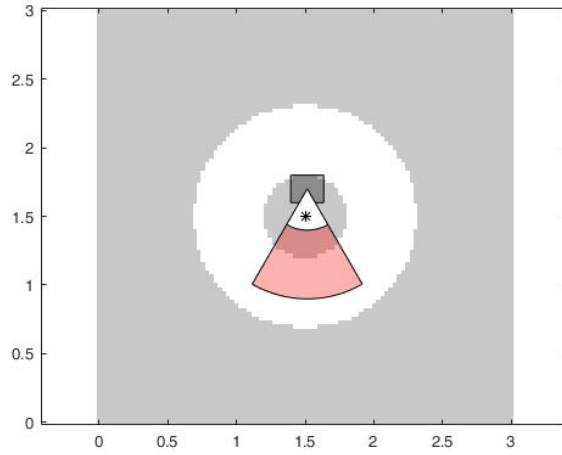
IV. $\mathcal{C}_{Target}$

Assume that the UGV camera is mounted at the center the UGV and always face the same direction as the UGV. $\mathcal{C}_{Target}$ is defined so that the target $\mathcal{T}_i$ in $\mathcal{W}$ maps in the robot's configuration space $\mathcal{C}_{free}$ to the C-target region $\mathcal{CT}_i = \{q \in \mathcal{C}_{free} | \mathcal{S}(q) \cap \mathcal{T}_i \neq \emptyset\}$. A simple $x - y$ plane demonstration (Figure 7) shows the effect of the distance to the target. In (a), the robot is within $\mathcal{C}_{Target}$, since the target is in the robot's FOV when it faces $-\pi/2$. In (b), the robot is not in $\mathcal{C}_{Target}$, and it can not detect the target no matter how it rotates at its current position.

Also, the visibility problem is introduced to deal with the case that the camera view is blocked by obstacles. That is, a target must be in the FOV of the given UGV configuration, and at the same time, if a line is drawn between the robot and the target, such a line must not cross any obstacle, demonstrated in (Figure 6). If this criteria is met, such a UGV configuration is defined as in the $\mathcal{C}_{target}$.

The C-Target is plotted for the workspace (Figure 9). A zoom-in example is shown in (Figure 10.
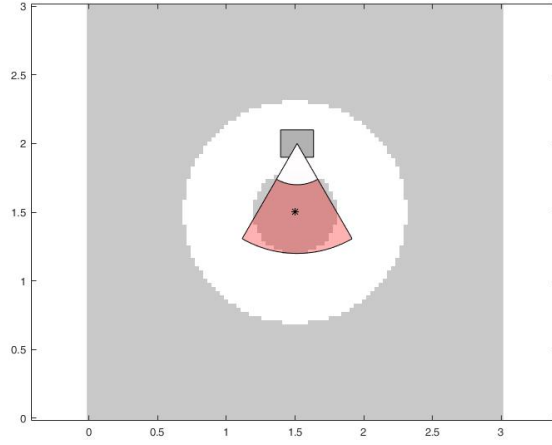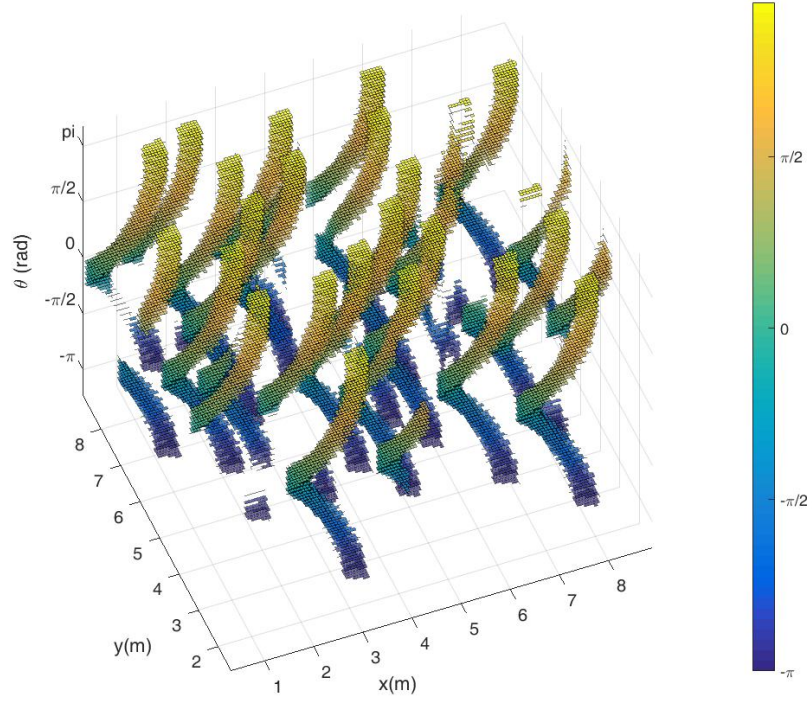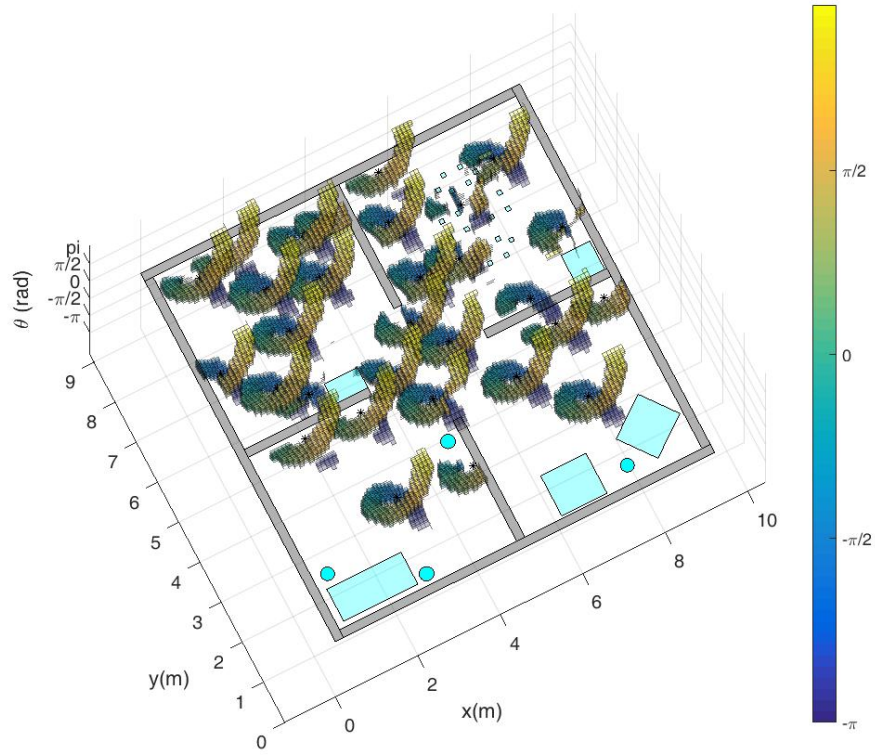
(a) robot is in C-target

(b) robot is not in C-target

Fig. 7: Example of C-target with one target. C-target denoted by white area, and the star denotes the target. The target must fall in the red region(robot FOV) to be classified or measured. In both examples, the UGV is facing $\theta = -\pi/2$
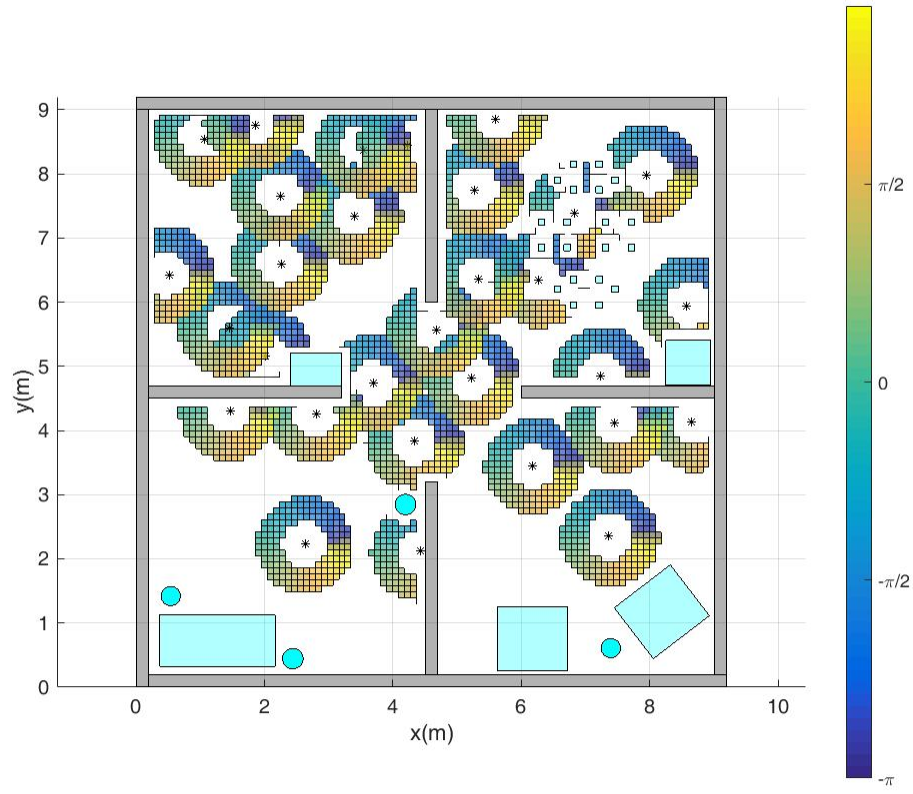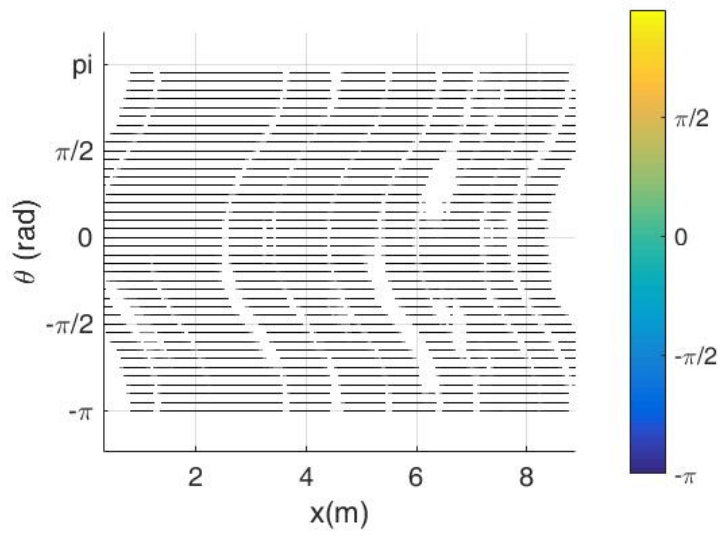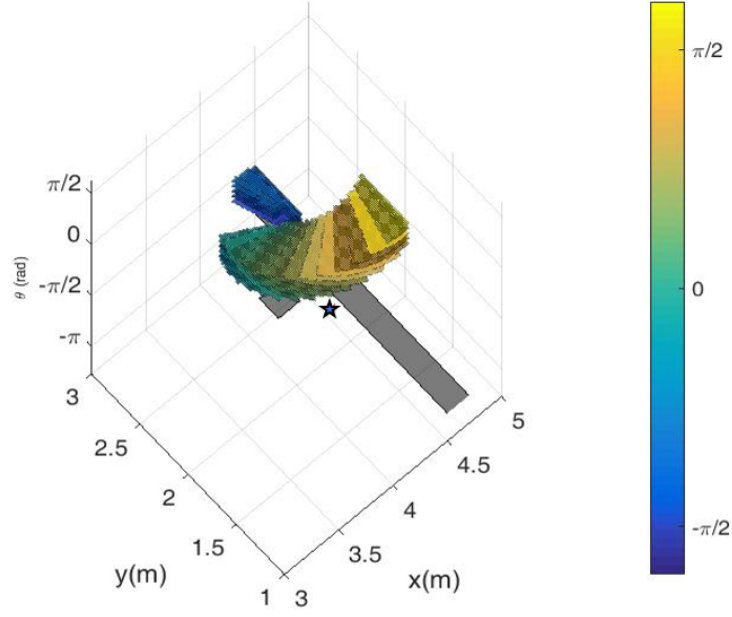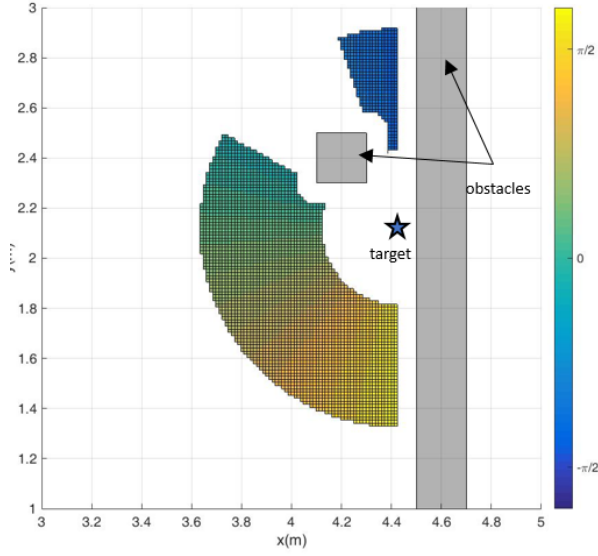
(a)

(b)

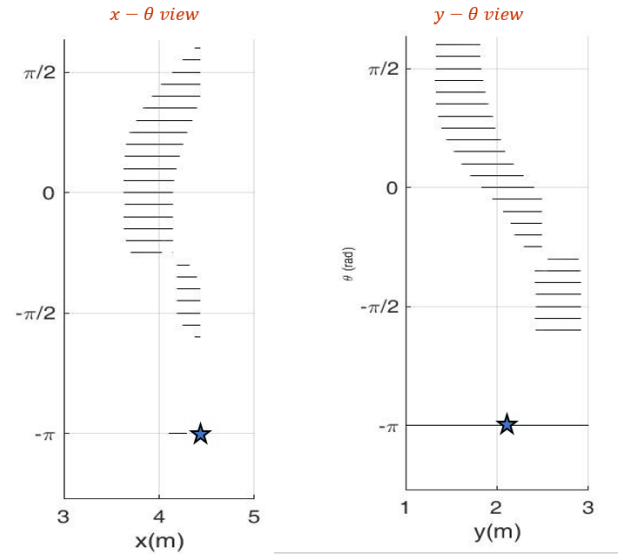Fig. 8: Perspective view of $\mathcal{C}_{target}$

(a) $x - y$ view

(b) $x - \theta$ view

Fig. 9: $\mathcal{C}_{target}$

(a) perspective view

(b) $x - y$ view

(c) $x - \theta$ view and $y - \theta$ view

Fig. 10: An example zoom-in of $\mathcal{C}_{target}$